

# Apéndice A

---

## Formalismo geométrico de la mecánica clásica

---

### A.1. Geometría diferencial

Comencemos con un preámbulo matemático en el que introducimos la definición de variedad diferenciable, que permite generalizar el concepto de diferenciabilidad en  $\mathbb{R}^n$ , esencial para cualquier teoría física, a espacios topológicos más generales.

**Definición A.1.** Sea  $M$  un espacio topológico Hausdorff y 2-numerable. Se dice que  $M$  es una **variedad diferenciable (sin borde)** si existe un conjunto de pares  $\{(U_i, \varphi_i)\}$  tal que

- (I)  $U_i$  es abierto de  $M$  y el conjunto de abiertos  $\{U_i\}$  define un recubrimiento de  $M$ .
- (II)  $\varphi_i: U_i \rightarrow V_i \subset \mathbb{R}^n$ , con  $V_i$  abierto, son aplicaciones biyectivas.
- (III) Las funciones transición  $\psi_{ij}: V_i \subset \mathbb{R}^n \rightarrow V_j \subset \mathbb{R}^n$  son de clase  $\mathcal{C}^\infty$ .

Dado un punto  $p \in M$ , la  $n$ -tupla  $\varphi_i(p) \in V_i \subset \mathbb{R}^n$  son las **coordenadas** de  $p$  en la carta  $(U_i, \varphi_i)$ . Al conjunto  $\{(U_i, \varphi_i)\}$  se le llama **atlas**  $\mathcal{A}$  de la variedad  $M$ .

**Definición A.2.** Sean  $M$  y  $N$  variedades diferenciables,  $F: M \rightarrow N$  una aplicación. Se dice que  $F$  es  $\mathcal{C}^p$ -diferenciable ( $1 \leq p \leq \infty$ ) si y sólo si todas las restricciones

$$F_{ij} = \varphi_j^N \circ F \circ (\varphi_i^M)^{-1}$$

son de clase  $\mathcal{C}^p$  en el sentido de  $\mathbb{R}^n$ . En particular, si se tiene una función  $f: M \rightarrow \mathbb{R}$ , se dice que es  $\mathcal{C}^p$ -diferenciable ( $1 \leq p \leq \infty$ ) si y sólo si todas las restricciones

$$F_i = F \circ (\varphi_i^M)^{-1}$$

son de clase  $\mathcal{C}^p$  en el sentido de  $\mathbb{R}^n$ . Consideraremos normalmente  $p = \infty$  en ambos casos.

Sabemos que el concepto de vector es un ingrediente básico de la descripción geométrica de un sistema dinámico en  $\mathbb{R}^n$  (vector tangente a una curva, velocidad de un sistema mecánico). Trasladamos este concepto a variedades diferenciables mediante los espacios tangentes.

**Definición A.3.** Sea  $M$  una variedad diferenciable y  $p \in M$ . Sea  $\mathcal{C}^\infty(p)$  el conjunto de funciones diferenciables definidas en un entorno abierto  $U$  de  $p$ . Considerando la operación bilineal

$$\begin{aligned}\cdot : \mathcal{C}^\infty(p) \times \mathcal{C}^\infty(p) &\rightarrow \mathcal{C}^\infty(p) \\ (f, g) &\mapsto \{q \mapsto (f \cdot g)(q) = f(q)g(q), q \in U\}\end{aligned}$$

se dice que  $(\mathcal{C}^\infty(p), \cdot)$  es un álgebra. Ahora, llamamos **vector tangente** en el punto  $p$  a cualquier derivación del álgebra, i.e. cualquier aplicación lineal

$$X_p : \mathcal{C}^\infty(p) \rightarrow \mathbb{R}$$

que satisface la regla de Leibniz:

$$X_p(f \cdot g) = X_p(f)g(p) + f(p)X_p(g).$$

El conjunto de vectores tangentes en el punto  $p \in M$  se llama **espacio tangente** de  $M$  en el punto  $p$ , denotado como  $T_p M$ . Tiene estructura de espacio vectorial.

Introducimos ahora la noción de campo vectorial:

**Definición A.4.** Sea  $M$  una variedad diferenciable y  $TM = \bigcup_{p \in M} T_p M$  su fibrado tangente. Un **campo vectorial**  $X$  es una aplicación

$$X : M \rightarrow TM$$

que asocia un vector a cada punto de la variedad, i.e.

$$X(p) = X_p \in T_p M.$$

El campo vectorial  $X$  es **diferenciable** cuando define una derivación del álgebra  $(\mathcal{C}^\infty(M), \cdot)$  en la forma:

$$D_X f(p) = X_p(f).$$

El conjunto de campos vectoriales diferenciables sobre la variedad  $M$  se denotará como  $\mathfrak{X}(M)$ .

**Definición A.5.** Llamaremos **curva integral** del campo vectorial  $X$  en el punto  $p \in M$  a la curva  $\sigma(t)$  en  $M$  cuyas coordenadas sobre una carta satisfacen

$$\frac{d\sigma^i(t)}{dt} = [X(\sigma(t))]^i.$$

La unicidad de solución nos permite definir un difeomorfismo

$$\Phi : \mathbb{R} \times M \rightarrow M$$

que asocia a cada punto de la variedad el punto alcanzado a tiempo  $t$  siguiendo la curva integral que empieza en el punto dado. Llamaremos a este difeomorfismo el **flujo** del campo vectorial  $X$ .

Los objetos duales a los campos vectoriales son las uno formas:

**Definición A.6.** Sea  $M$  una variedad diferenciable. Llamamos **uno forma** a una aplicación

$$\alpha: M \rightarrow T^*M = \bigcup_p T_p^*M$$

tal que asocia un covector a cada punto de la variedad, i.e.

$$\alpha(p) \in T_p^*M \quad \forall p \in M.$$

El conjunto de uno formas sobre la variedad  $M$  se denotará  $\Lambda^1(M)$ .

La construcción realizada para campos y uno formas se puede generalizar para campos tensoriales arbitrarios  $r$  veces covariantes y  $s$  contravariantes. Se construyen así aplicaciones

$$\tau: M \rightarrow T^{r,s}M = \bigcup_p (\tau_p)_s^r$$

que asignan a cada punto de la variedad un tensor de los órdenes adecuados.

**Definición A.7.** Sea  $M$  una variedad diferenciable. Llamamos **r-formas (diferenciables)** a cualquier tensor  $(0, r)$  completamente antisimétrico. El conjunto de r-formas se puede denotar  $\Lambda^r(M)$ . El conjunto de formas de todos los órdenes se define como

$$\Lambda(M) = \bigoplus_k \Lambda^k(M).$$

Sobre este conjunto se puede definir un operador, la **diferencial exterior**, que actúa sobre las funciones como:

$$d: C^\infty(M) \rightarrow \Lambda^1(M)$$

tal que

$$df(X) = X(f).$$

Además, se puede extender su acción al conjunto  $\Lambda(M)$  imponiendo que:

- Si  $\omega \in \Lambda^r(M)$  y  $\alpha \in \Lambda^s(M)$ , entonces

$$d(\omega \wedge \alpha) = d\omega \wedge \alpha + (-1)^r \omega \wedge d\alpha.$$

- $d$  es nilpotente, i.e.

$$d^2 = 0.$$

## A.2. Dinámica clásica

Sea un sistema clásico con  $n$  grados de libertad. Su espacio de fases es  $M_C$ , una variedad diferenciable  $2n$ -dimensional (localmente homeomorfa a  $\mathbb{R}^{2n}$ ). Las  $2n$  dimensiones se corresponden con  $n$  coordenadas de posición que especifican la configuración del sistema y los  $n$  momentos correspondientes.

Los observables en  $M_C$  son funciones diferenciables

$$f: M_C \longrightarrow \mathbb{R}$$

que asignan a cada punto del espacio de fases el resultado de una medida. En este conjunto de funciones  $\mathcal{C}^\infty(M_C)$  se introduce la siguiente operación:

**Definición A.8.** Se define el **paréntesis de Poisson**,  $\{\cdot, \cdot\}$  como la operación bilineal

$$\{\cdot, \cdot\}: \mathcal{C}^\infty(M_C) \times \mathcal{C}^\infty(M_C) \longrightarrow \mathcal{C}^\infty(M_C)$$

que es antisimétrica, satisface la identidad de Jacobi y la regla de Leibniz como en 2.2. El par  $(\mathcal{C}^\infty(M_C), \{\cdot, \cdot\})$  se dice **álgebra de Poisson** de  $M_C$ , que se convierte en una **variedad de Poisson**.

Un caso particular de variedades de Poisson son las no degeneradas, i.e. aquellas para las que el conjunto de funciones de Casimir es vacío. Veamos primero qué es una función de Casimir:

**Definición A.9.** Sea  $(M, \{\cdot, \cdot\})$  una variedad de Poisson. Se llama **función de Casimir** a cualquier función  $C \in \mathcal{C}^\infty(M)$  tal que el paréntesis de Poisson aplicado con cualquier otra función del álgebra se anula, i.e.

$$\{C, f\} = 0 \quad \forall f \in \mathcal{C}^\infty(M).$$

Introduzcamos ahora un tipo de variedades de Poisson particularmente interesante, las variedades simplécticas:

**Definición A.10.** Sea  $M_C$  una variedad diferenciable y sea  $\omega \in \Lambda^2(M_C)$  una dos forma. Diremos que  $\omega$  es una **forma simpléctica** si y sólo si

- Es cerrada, i.e.

$$d\omega = 0.$$

- Es no degenerada, i.e. dado un campo vectorial  $X \in \mathfrak{X}(M_C)$  la relación

$$\omega(X, Y) = 0 \quad \forall Y \in \mathfrak{X}(M_C) \implies X = 0.$$

Llamamos al par  $(M_C, \omega)$  **variedad simpléctica**.

Se puede probar que si una variedad es de Poisson no degenerada, entonces es simpléctica. En este caso el teorema de Darboux asegura que existe un conjunto de coordenadas canónicas para las que el paréntesis tiene la forma estándar.

**Teorema A.11** (Darboux). *Sea  $M_C$  una variedad simpléctica. Entonces, existe un atlas en  $M_C$  tal que las coordenadas de un punto son  $(R^k, P_k)$ , las **coordenadas de Darboux**, y localmente se verifica*

$$\{f_1, f_2\} = \sum_{k=1}^n \frac{\partial f_1}{\partial P_k} \frac{\partial f_2}{\partial R^k} - \frac{\partial f_1}{\partial R^k} \frac{\partial f_2}{\partial P_k}.$$

El paréntesis de Poisson nos permite introducir los conceptos de campo vectorial Hamiltoniano y sistema Hamiltoniano:

**Definición A.12.** Sea una variedad de Poisson  $M_C$  y una función  $f \in \mathcal{C}^\infty(M_C)$ . Un campo vectorial  $X_f$  se dice que es el **campo vectorial Hamiltoniano** de  $f$  si

$$X_f(g) = \{f, g\} \quad \forall g \in \mathcal{C}^\infty(M_C).$$

En las coordenadas de Darboux toma la forma:

$$X_f = \sum_{k=1}^n \frac{\partial f(R, P)}{\partial P_k} \frac{\partial}{\partial R^k} - \frac{\partial f(R, P)}{\partial R^k} \frac{\partial}{\partial P_k}.$$

**Definición A.13.** Llamamos **sistema Hamiltoniano** a la terna  $(M_C, \{\cdot, \cdot\}, H)$ , donde  $\{\cdot, \cdot\}$  es el paréntesis de Poisson sobre  $M_C$  y  $H \in \mathcal{C}^\infty(M_C)$  es el Hamiltoniano del sistema.

La dinámica de un sistema Hamiltoniano puede formularse de dos formas alternativas:

- Las trayectorias del sistema vienen dadas por las curvas integrales del campo vectorial Hamiltoniano,  $X_H$ .
- Si se considera el conjunto de observables, la dinámica se escribe mediante la actuación del paréntesis de Poisson del Hamiltoniano con cualquier otra función,

$$\frac{df}{dt} = \{H, f\} \quad \forall f \in \mathcal{C}^\infty(M_C). \quad (\text{A.1})$$

Claramente son perspectivas equivalentes, puesto que las ecuaciones diferenciales que determinan las curvas integrales del campo vectorial Hamiltoniano vienen dadas por la ecuación (A.1) para las funciones ‘posición’ y ‘momento’ de cada partícula ( $R$  y  $P$ ), que no son otra cosa que las ecuaciones de Hamilton

$$\dot{R}^j = \frac{\partial H}{\partial P_j} \quad \dot{P}_j = -\frac{\partial H}{\partial R^j}$$

Finalizamos este apéndice con un teorema que cobra gran relevancia en mecánica estadística:

**Teorema A.14** (Liouville). *Sea  $(M, \omega, H)$  un sistema Hamiltoniano sobre una variedad simpléctica y  $\Phi_t: M \rightarrow M$  el difeomorfismo asociado al flujo del campo Hamiltoniano  $X_H$ . Entonces,  $\Phi_t$  es un simplectomorfismo, es decir, preserva la forma simpléctica*

$$\Phi_t^* \omega = \omega.$$

*En particular, también preserva el elemento de volumen infinitesimal  $\Omega = \frac{1}{n!} \omega^{\wedge n}$ , lo que corresponde al teorema de Liouville clásico para sistemas Hamiltonianos.*

## Apéndice B

### Formalismo geométrico de la mecánica cuántica

El objetivo de esta sección es proporcionar una descripción de los sistemas cuánticos empleando las mismas herramientas geométricas de las que se ha hecho uso en el anexo A para describir los sistemas mecánicos clásicos. En este trabajo se va a hacer únicamente una revisión breve del marco teórico, para un enfoque más detallado consultar [5].

#### B.1. Dinámica cuántica en términos de estados

Por simplicidad, nos restringiremos al caso finito-dimensional. El espacio de Hilbert que actúa como espacio de fases del sistema es isomorfo a  $\mathbb{C}^n$  para  $n \in \mathbb{N}$ ,  $\mathcal{H} \equiv \mathbb{C}^n$ .

##### B.1.1. Los estados

Sea una base  $\{|\psi_k\rangle\}$  de  $\mathcal{H}$ . Cada estado  $|\psi\rangle \in \mathcal{H}$  puede expresarse como combinación lineal de elementos de la base con coeficientes complejos  $\{z_k\}$ :

$$|\psi\rangle = \sum_k z_k |\psi_k\rangle.$$

Podemos tomar el espacio vectorial inherente al espacio de Hilbert y considerarlo un espacio vectorial real  $M_Q$ , separando cada coordenada en sus partes real e imaginaria:

$$z_k = q_k + ip_k \mapsto (q_k, p_k) \in \mathbb{R}^{2n} \equiv M_Q.$$

Emplearemos coordenadas reales  $(q_k, p_k)$ ,  $k = 1, \dots, n$ , para representar los puntos de  $\mathcal{H}$  entendidos como puntos de una variedad real. Es preciso tener presente que esta notación no tiene que ver con las coordenadas clásicas usuales, es puramente cuántica. Tras esta realificación, las similitudes entre la dinámica cuántica y la clásica descrita en la sección anterior resultarán más evidentes.

Notemos en primer lugar que no existe una correspondencia biyectiva entre un estado físico y los elementos del espacio de Hilbert que lo describen; es un hecho bien conocido que, debido a los postulados de medida de la mecánica cuántica, la fase global no aporta información física. Esta ambigüedad se tiene que traducir por lo tanto a la nueva formulación en términos de espacios vectoriales reales. Para ello, dada la relación de equivalencia en  $\mathcal{H}$

$$|\psi\rangle \sim |\varphi\rangle \iff |\varphi\rangle = e^{i\theta} |\psi\rangle, \theta \in \mathbb{R}$$

la trasladamos a  $M_Q$  observando que, en coordenadas polares  $(\{r_k, \theta_k\}_{k=1, \dots, n} \text{ con } z_k = r_k e^{i\theta_k})$ , dos es-

tados equivalentes están relacionados por una transformación de fase, es decir, pertenecen a la misma órbita del flujo generado por el campo  $\Gamma = \sum_k \partial \theta_k$ , que en coordenadas cartesianas toma la forma

$$\Gamma := \sum_k \left( q_k \frac{\partial}{\partial p_k} - p_k \frac{\partial}{\partial q_k} \right). \quad (\text{B.1})$$

Podemos interpretar geométricamente el campo (B.1) de dos formas:

- calculando sus curvas integrales, que son los diferentes estados que se obtienen a partir de multiplicar por una fase global uno dado;
- actuando con el campo vectorial sobre las funciones de  $M_Q$ . Una función  $f \in \mathcal{C}^\infty(M_Q)$  será un observable válido si es invariante bajo transformaciones de fase, i.e.

$$\mathcal{L}_\Gamma f = \Gamma f = 0, \quad (\text{B.2})$$

aunque no será la única condición.

Existe además otra relación de equivalencia en un espacio de fases cuántico: consideramos dos estados equivalentes si difieren únicamente en norma. Por tanto, podemos considerar únicamente la esfera de estados de norma 1 en  $\mathbb{C}^n$ :

$$S_Q := \left\{ (q, p) \in M_Q \mid \sum_k (q_k^2 + p_k^2) = 1 \right\}. \quad (\text{B.3})$$

Es inmediato ver que el campo vectorial  $\Gamma$  es tangente a  $S_Q$ , dado que el cambio de fase preserva la norma.

### B.1.2. El paréntesis de Poisson

Veamos que se puede definir un paréntesis de Poisson en  $M_Q$  y de la definición deduciremos una importante propiedad.

**Definición B.1.** Sea  $M_Q$  la realificación del espacio de Hilbert  $\mathcal{H} \sim \mathbb{C}^n$ , con coordenadas  $z_k = q_k + ip_k$  las componentes de un vector en base ortonormal  $\{|\psi_k\rangle\}$ , sean  $f, g \in \mathcal{C}^\infty(M_Q)$ , se define el **paréntesis de Poisson** en  $M_Q$  como

$$\{f, g\} := \sum_k \frac{1}{2} \left( \frac{\partial f}{\partial p_k} \frac{\partial g}{\partial q_k} - \frac{\partial f}{\partial q_k} \frac{\partial g}{\partial p_k} \right).$$

Será útil introducir también un paréntesis simétrico:

$$\{f, g\}_+ := \sum_k \frac{1}{2} \left( \frac{\partial f}{\partial q_k} \frac{\partial g}{\partial q_k} + \frac{\partial f}{\partial p_k} \frac{\partial g}{\partial p_k} \right).$$

**Lema B.2.** Sean  $f, g \in \mathcal{C}^\infty(M_Q)$  posibles observables cuánticos (i.e.  $\Gamma f = \Gamma g = 0$ ). Entonces,  $\{f, g\}$  y  $\{f, g\}_+$  verifican también  $\Gamma\{f, g\} = \Gamma\{f, g\}_+ = 0$ .

*Demostración.* Basta probar que  $\Gamma\{f, g\} = \{\Gamma f, g\} + \{f, \Gamma g\}$  y  $\Gamma\{f, g\}_+ = \{\Gamma f, g\}_+ + \{f, \Gamma g\}_+$  mediante un cálculo directo.  $\square$

### B.1.3. Los observables

Procedemos ahora a discutir cómo representar los observables físicos en este nuevo marco. En lugar de representar los observables como operadores lineales autoadjuntos sobre  $\mathcal{H}$ , los representamos como funciones definidas sobre la variedad real  $M_Q$ , al igual que se hizo en MC. No obstante, debido a la linealidad de los operadores, no toda función de  $C^\infty(M_Q)$  es válida. Consideraremos para todo operador  $A \in \text{Lin}(\mathcal{H})$ , la función cuadrática asociada

$$f_A := \langle \psi | A \psi \rangle. \quad (\text{B.4})$$

Denotemos el conjunto de tales funciones por  $\mathcal{F}_s(M_Q)$ . Notemos que, a diferencia del caso clásico en el que cada punto de  $M_C$  proporciona un resultado bien definido para un observable clásico  $f: M_C \rightarrow \mathbb{R}$ , en mecánica cuántica geométrica (MQG) un observable  $f_A \in \mathcal{F}_s(M_Q)$  evaluado sobre el estado  $|\psi\rangle$  proporciona el valor esperado del operador  $A$  sobre el estado, de manera que la función asociada de la forma (B.4) está bien definida.

Una vez propuesta la definición es preciso comprobar que es consistente, es decir, que se cumple la condición (B.2) y además que se preservan las estructuras algebraicas que existen en el conjunto de operadores en MQ usual. Existen tres de estas estructuras significativas para la descripción física:

- El producto asociativo de dos operadores

$$A, B \in \mathcal{H} \longrightarrow A \cdot B \in \text{Lin}(\mathcal{H}).$$

Es importante observar que esta operación no es interna en el conjunto de operadores Hermíticos (aquellos asociados con magnitudes físicas), dado que el producto de dos operadores Hermíticos no lo es en general.

- El anticomutador de dos operadores

$$A, B \in \text{Lin}(\mathcal{H}) \longrightarrow [A, B]_+ \in \text{Lin}(\mathcal{H}).$$

- El comutador de dos operadores

$$A, B \in \text{Lin}(\mathcal{H}) \longrightarrow i[A, B] \in \text{Lin}(\mathcal{H}).$$

Las dos últimas operaciones sí son internas en el espacio de operadores Hermíticos.

Veamos finalmente el resultado de consistencia que sintetiza la construcción realizada hasta ahora.

**Teorema B.3.** *Sea  $\mathcal{H} \sim \mathbb{C}^n$  un espacio de Hilbert con base ortonormal  $\{|\psi_k\rangle\}$ ,  $M_Q$  la variedad real asociada. Para cada observable cuántico  $A \in \text{Lin}(\mathcal{H})$  se define una función  $f_A \in C^\infty(M_Q)$ ,  $f_A := \langle \psi | A \psi \rangle$  y denotamos el conjunto de tales funciones por  $\mathcal{F}_s(M_Q)$ . Además, esta función satisface:*

- (I) *Está físicamente bien definida, i.e.  $\Gamma f_A = 0$ .*
- (II) *Esta identificación define un isomorfismo para las las estructuras algebraicas definidas en  $\text{Lin}(\mathcal{H})$ , en particular:*
  - $[A, B]_+ \longrightarrow f_{[A, B]_+} = \{f_A, f_B\}_+ \in \mathcal{F}_s(M_Q)$ ,
  - $i[A, B] \longrightarrow f_{i[A, B]} = \{f_A, f_B\} \in \mathcal{F}_s(M_Q)$ .

*Demostración.* Se sigue por construcción que  $f_A$  tiene derivada de Lie nula sobre el campo  $\Gamma$  de cambio de fase global. Comprobar la segunda propiedad es costoso y remitimos a [1].  $\square$

Concluimos que los operadores cuánticos y sus estructuras algebraicas asociadas quedan bien codificados con la identificación (B.4) y que  $\mathcal{F}_s(M_Q)$  es cerrado respecto a los paréntesis de Poisson de la definición B.1.

#### B.1.4. La dinámica

Análogamente al caso clásico, la dinámica se puede construir de varias formas compatibles con las estructuras geométricas introducidas hasta ahora.

- En la interpretación de Schrödinger de la MQ la dinámica se describe como las curvas integrales del campo vectorial  $X_{f_H}$ , donde  $H$  es el operador Hamiltoniano del sistema:

$$X_{f_H} = \hbar^{-1} \{ f_H, \cdot \}.$$

- En la interpretación de Heisenberg, la dinámica se introduce trasladando la ecuación de Heisenberg  $\partial_t A = -i\hbar^{-1}[A, H] = \hbar^{-1}(i[H, A])$  a términos geométricos aplicando el teorema B.3:

$$\dot{f}_A = \hbar^{-1} \{ f_H, f_A \}.$$

Por lo tanto, es posible describir la dinámica cuántica como el flujo de un campo vectorial Hamiltoniano en la variedad real  $M_Q$  o bien en el conjunto de observables  $\mathcal{F}_s(M_Q)$ . Ahora, como se introdujo en la sección B.1.1, en el espacio de fases cuántico podemos restringirnos a la esfera de norma unidad  $S_Q$  sin perder información física. Es por tanto relevante el siguiente resultado:

**Proposición B.4.** *La dinámica definida sobre  $M_Q$  por  $X_{f_H}$  preserva la norma del estado. Por tanto, el flujo está restringido a la esfera  $S_Q$ .*

*Demostración.* Por construcción, el operador  $\mathbb{I} \in \text{Lin}(\mathcal{H})$  tiene como función asociada la norma del estado

$$f_{\mathbb{I}}(q_k, p_k) = \sum_k (q_k^2 + p_k^2), \quad f_{\mathbb{I}} \in \mathcal{F}_s(M_Q).$$

Su evolución por el flujo Hamiltoniano vendrá dada por

$$\dot{f}_{\mathbb{I}} = \hbar^{-1} \{ f_H, f_{\mathbb{I}} \}.$$

Finalmente, aplicando el teorema B.3,

$$\{ f_H, f_{\mathbb{I}} \} = f_{i[H, \mathbb{I}]} = 0 \implies \dot{f}_{\mathbb{I}} = 0.$$

$\square$

## B.2. Dinámica cuántica en términos de proyectores

La siguiente discusión se puede encontrar en [12] y [13].

### B.2.1. Los estados

A lo largo de la construcción de la MQ geométrica hemos mencionado que los estados físicos están representados por los rayos de  $\mathcal{H}$ . Esto es, si consideramos  $|\psi\rangle \in \mathcal{H} \setminus \{0\} = \hat{\mathcal{H}}$ , cualquier  $re^{i\theta}|\psi\rangle$  con  $r > 0$ ,  $\theta \in \mathbb{R}$ , codifica la misma información medible. Consideraremos un nuevo espacio de estados, que si se tenía el isomorfismo  $\mathcal{H} \equiv \mathbb{C}^n$ , será isomorfo al espacio proyectivo  $\mathbb{CP}^{n-1}$ .

El siguiente paso es traducirlo a términos geométricos. La manera de hacerlo es introduciendo una relación de equivalencia  $\mathcal{R}$  en  $\hat{M}_Q$  (que denota la estructura de variedad de  $\hat{\mathcal{H}}$ ). Para ello se introducen los campos vectoriales

$$\Delta = \sum_{k=1}^n q_k \frac{\partial}{\partial q_k} + p_k \frac{\partial}{\partial p_k}, \quad \Gamma = \sum_{k=1}^n p_k \frac{\partial}{\partial q_k} - q_k \frac{\partial}{\partial p_k}$$

que geométricamente representan las variaciones de módulo y de fase global de  $|\psi\rangle$  respectivamente. Efectivamente, en coordenadas polares,

$$\Delta = \sum_{k=1}^n r_k \frac{\partial}{\partial r_k}, \quad \Gamma = \sum_{k=1}^n \theta_k \frac{\partial}{\partial \theta_k}.$$

Así, podemos definir la relación de equivalencia  $\mathcal{R}$ :

$$x, y \in \hat{M}_Q, x \mathcal{R} y \iff \begin{array}{l} x \text{ e } y \text{ pertenecen a la misma órbita generada} \\ \text{por la acción de algún } X \in \mathbb{R}\langle\Delta, \Gamma\rangle \text{ sobre } \hat{M}_Q \end{array}$$

Como  $[\Delta, \Gamma] = 0$  la distribución  $\{\Delta, \Gamma\}$  es integrable por el teorema de Fröbenius y define una foliación de  $\hat{M}_Q$ . Así, el conjunto  $\hat{M}_Q/\mathcal{R}$  tiene estructura de variedad cociente. Este conjunto, que denominaremos por  $\mathcal{P}$ , es difeomorfo al espacio proyectivo complejo  $\mathbb{CP}^{n-1}$ . Podemos definir la proyección canónica

$$\pi: \hat{M}_Q \rightarrow \mathcal{P}, |\psi\rangle \mapsto [\psi]$$

donde  $[\psi]$  es la clase de equivalencia con representante el punto de  $M_Q$  que se asocia a un  $|\psi\rangle \in \hat{\mathcal{H}}$ .

### B.2.2. Los observables

Redefinimos ahora el espacio de observables  $\mathcal{F}_s(M_Q)$ . El nuevo espacio de funciones que utilizaremos es

$$\mathcal{F}_{\mathcal{O}}(\hat{M}_Q) = \left\{ e_A = \frac{\langle \psi | A \psi \rangle}{\langle \psi | \psi \rangle} \mid |\psi\rangle \in \mathcal{H}, A \in \text{Herm}(\mathcal{H}) \right\}.$$

Por el teorema B.3 y por construcción, los observables de  $\mathcal{F}_{\mathcal{O}}$  son constantes en las curvas integrales de  $\Delta$  y  $\Gamma$ , permitiendo así construir un conjunto de funciones bien definidas sobre la variedad cociente  $\mathcal{P}$ . Finalizamos la discusión relativa a los operadores en MQG con la teoría espectral, de máxima importancia en QM. En el marco geométrico, se consideran los autovalores y autoestados de la siguiente forma:

**Teorema B.5.** Sea  $e_A(\psi) = \frac{f_A(\psi)}{\langle \psi | \psi \rangle} \in \mathcal{F}_s(M_Q)$  la función asociada al observable  $A$ . Si consideramos  $\psi \in S_Q$ ,

(I) los autoestados del operador  $A$  coinciden con los puntos críticos de la función  $f_A$ , i.e.  $df_A(\psi) = 0 \implies \psi$  es un autoestado de  $A$ .

(II) el autovalor de  $A$  en el autovalor  $\psi$  es el valor que toma la función  $f_A$  en el punto crítico.

### B.2.3. La aplicación momento

Dada la acción de un grupo sobre una variedad, en particular  $U(n) \times \mathcal{P} \rightarrow \mathcal{P}$  se puede definir entre la variedad y el dual al álgebra de Lie la aplicación momento. En este caso, define un difeomorfismo entre el espacio proyectivo  $\mathbb{CP}^{n-1}$  (equivalentemente  $\mathcal{P}$  con estructura de variedad) y el subespacio de proyectores de rango uno de  $\text{Herm}(\mathcal{H})$ , que denotaremos  $\mathcal{D}^1(\mathcal{H})$

$$\begin{aligned}\mu: \mathcal{P} &\longrightarrow \mathcal{D}^1(\mathcal{H}) \subset \mathfrak{u}^*(\mathcal{H}) \\ [\psi] &\longmapsto \mu([\psi]) = \frac{|\psi\rangle\langle\psi|}{\langle\psi|\psi\rangle} = \rho_\psi\end{aligned}$$

donde denotamos por  $\mathfrak{u}(\mathcal{H}) \equiv \mathfrak{u}(n)$  el álgebra de Lie del grupo unitario  $U(n)$ , y por  $\mathfrak{u}^*(\mathcal{H}) \equiv \mathfrak{u}^*(n)$  su dual.

Los elementos de  $\mathcal{D}^1(\mathcal{H})$ , que se corresponden<sup>1</sup> con las matrices densidad de estados puros, se pueden escribir con coordenadas reales en una base del espacio de operadores correspondiente al ser isomorfo a un subespacio de  $\text{Herm}(\mathcal{H})$ .

Sobre el dual  $\mathfrak{g}^*$  a un álgebra de Lie  $(\mathfrak{g}, [\cdot, \cdot])$ , se puede construir un paréntesis de Lie-Poisson  $\{\cdot, \cdot\}$  en la forma

$$\{\cdot, \cdot\} = \sum_{j,k,l} c_{jk}^l x_l \frac{\partial}{\partial x^j} \otimes \frac{\partial}{\partial x^k}$$

siendo  $c_{jk}^l$  las constantes de estructura del álgebra de Lie para una base  $\{A_j\}$  de  $\mathfrak{g}$ ,

$$[A_j, A_k] = c_{jk}^l A^l$$

y  $x_l$  las coordenadas en  $\mathfrak{g}^*$  respecto a la base dual. En particular, para  $(\mathfrak{u}(n), [\cdot, \cdot])$  podemos dotar a su dual y por tanto a su subespacio  $\mathcal{D}^1(\mathcal{H})$  de estructura de Poisson mediante

$$\{f_A, f_B\}_Q = \sum_{j,k,l} C_{jk}^l x_l \frac{\partial f_A}{\partial x^j} \frac{\partial f_B}{\partial x^k} \tag{B.5}$$

donde  $\{x_1, \dots, x_{n^2}\}$  son las coordenadas del dual a  $\text{Herm}(n)$  consideradas respecto a su base  $\{\sigma^k\}$ , con las constantes de estructura del álgebra de Lie

$$[\sigma_j, \sigma_k] = C_{jk}^l \sigma_l$$

---

<sup>1</sup>Realmente son objetos duales, pero están relacionados por el isomorfismo canónico inducido por el producto escalar de la traza y el isomorfismo  $\mathfrak{u}(n) \rightarrow \text{Herm}(n)$ ,  $A \mapsto iA$ .

### B.2.4. La dinámica

Ahora, en el formalismo de matriz densidad cuántica, se tiene que la evolución temporal y el valor medio de un operador  $A$  vienen dados por

$$i\hbar \frac{d\rho(t)}{dt} = [H, \rho(t)] \quad (\text{B.6})$$

$$\langle A \rangle = \text{Tr}(\rho A) \quad (\text{B.7})$$

donde (B.6) es la ecuación de von Neumann. Estamos en condiciones de trasladarlo a términos geométricos:

$$i\hbar \frac{d\rho}{dt} = \{H, \rho\}_Q.$$

Empleando el isomorfismo canónico dado por el producto escalar de la traza y el hecho de que el espacio de operadores es real,  $\text{Herm}(\mathcal{H}) = \text{Herm}(n) = \mathbb{R}\langle \sigma_k \rangle$  para cierta base de operadores  $\sigma_k$  (en particular las matrices de Pauli en dimensión dos), podemos escribir el valor medio y la evolución temporal como:

$$\langle A \rangle = \text{Tr}(\rho A) = \sum_{i=1}^{n^2} \sum_{j=1}^{n^2} \rho_i A_j \text{Tr}(\sigma_i \sigma_j) = \sum_{i=1}^{n^2} \sum_{j=1}^{n^2} \rho_i A_j \delta_j^i = \sum_{i=1}^{n^2} A_i \rho_i \quad (\text{B.8})$$

$$i\hbar \frac{d\rho_k}{dt} = \{H, \rho_k\}_Q = \sum_{j=1, l=1}^{n^2} C_{jk}^l H_j \rho_l \quad (\text{B.9})$$

donde  $A_k, H_k$  y  $\rho_k$  son las coordenadas  $k$ -ésimas del operador  $A$ , del Hamiltoniano y de la matriz densidad en base  $\{\sigma_j\}$  respectivamente; y  $C_{jk}^l$  son las constantes de estructura del álgebra de Lie que vienen definidas por la relación  $[\sigma_j, \sigma_k] = C_{jk}^l \sigma_l$ .

# Apéndice C

---

## Resultados auxiliares

---

### C.1. Demostraciones

#### Teorema 2.4.

*Demostración.* Basta observar que para el operador evolución temporal  $\phi_t^*$  inducido por la dinámica Hamiltoniana se tiene que

$$\int_M dq dp \phi_t^*[F(q, p)] \phi_t^*[f(q, p)] = \int_M dq dp F(q, p) f(q, p)$$

aplicando el teorema de Liouville (A.14). Ahora, por (2.14)

$$\langle f(t) \rangle = \int_M dq dp F(q, p) \phi_t^*[f(q, p)]$$

y por la primera observación, podemos aplicar  $\phi_{-t}^*$  a ambas funciones sin cambiar el resultado de la integral,

$$\langle f(t) \rangle = \int_M dq dp \phi_{-t}^*[F(q, p)] \phi_{-t}^* \phi_t^*[f(q, p)] = \int_M dq dp \phi_{-t}^*[F(q, p)] f(q, p).$$

□

#### Teorema 2.7.

*Demostración.* En primer lugar, por la linealidad de la traza y su propiedad cíclica se tiene:

$$\langle A \rangle(R, P) := \text{Tr}[\rho(R, P)A(R, P)] = \int_{\mathcal{S}_Q} d\mu_Q(q, p) F_H(R, P, q, p) f_A(R, P, q, p).$$

Aplicando la definición 2.6, para obtener el promedio total correspondiente hay que integrar el objeto anterior en el espacio de fases clásico:

$$\langle A \rangle := \int_{M_C} d\mu_C(R, P) \langle A \rangle(R, P) = \int_{M_C} d\mu_C(R, P) \text{Tr}[\rho(R, P)A(R, P)].$$

□

**Teorema 3.4.**

*Demostración (idea).* Se trata de un problema de optimización restringida: encontrar la *matriz densidad híbrida* que maximiza  $S$  en la ecuación (3.4) sujeta a las siguientes restricciones:

$$\begin{aligned} C_N[\rho(\xi)] &:= \int_{M_C} d\mu_C(\xi) \text{Tr}[\rho(\xi)] - 1 = 0 \\ C_E[\rho(\xi)] &:= \int_{M_C} d\mu_C(\xi) \text{Tr}[\rho(\xi)H(\xi)] - E = 0. \end{aligned}$$

Estas restricciones pueden incorporarse mediante multiplicadores de Lagrange, definiendo el funcional a optimizar:

$$\mathcal{S} := S - \lambda_N C_N - \lambda_E C_E.$$

El resto de la prueba puede encontrarse en [2]. □

## C.2. Teorema de Gleason

**Teorema C.1** (Gleason). *Sea  $\mu$  una medida en los subespacios cerrados de un espacio (real o complejo) de Hilbert separable  $\mathcal{H}$  de dimensión al menos tres. Existe un operador autoadjunto semidefinito positivo  $T$  tal que para todo subespacio cerrado  $A$  de  $\mathcal{H}$*

$$\mu(A) = \text{Tr}(TP_A)$$

donde  $P_A$  es la proyección ortogonal de  $\mathcal{H}$  sobre  $A$ .

## C.3. Demostración del lema 2

Se quiere probar la ecuación (4.5),

$$\begin{aligned} \dot{\rho}(\xi) &= \left( -\frac{i}{\hbar} [H(\xi), \rho(\xi)] + \sum_k \lambda_k(\xi) \text{Tr}(\hat{\pi}_k(\xi) \{\xi, H(\xi)\}_C) \hat{\pi}'_k(\xi) \right) + \\ &\quad + \sum_k \text{Tr}(\{\lambda_k(\xi) \hat{\pi}_k(\xi), H(\xi)\}_C) \hat{\pi}_k(\xi) \end{aligned}$$

Dada  $\mathcal{F}$ , que actúa sobre  $A(\xi, \rho_\psi)$  en la forma,

$$\int_{M_H} A(\xi, \rho_\psi) \left( \sum_k \lambda_k(\xi) \delta(\rho_\psi - \hat{\pi}_k(\xi)) \right) d\mu_H =: \int_{M_H} A(\xi, \rho_\psi) \sum_k F_H^k(\xi, \rho_\psi),$$

queremos calcular  $\int_{\mathcal{D}^1(\mathcal{H})} d\mu_Q \dot{F}_H^k(\xi, \rho_\psi) \rho_\psi = \int d\mu_Q \{F_H^k, \text{Tr}(H(\xi) \rho_\psi)\} \rho_\psi$ . En el integrando aparecen tres términos (que denotamos por comodidad fuera de la integral pero sólo tienen sentido integrados),

$$\begin{aligned} &\left\{ \lambda_k(\xi), \text{Tr}[H(\xi) \rho_\psi] \right\}_C \delta(\rho_\psi - \hat{\pi}_k(\xi)) + \lambda_k(\xi) \left\{ \delta(\rho_\psi - \hat{\pi}_k(\xi)), \text{Tr}[H(\xi) \rho_\psi] \right\}_C + \\ &\quad + \lambda_k(\xi) \left\{ \delta(\rho_\psi - \hat{\pi}_k(\xi)), \text{Tr}[H(\xi) \rho_\psi] \right\}_Q \end{aligned}$$

Integrando estos tres términos pesados con  $\rho_\psi$  en el espacio proyectivo y sumando a  $k$ , se tiene la ecuación buscada (tras reordenar y agrupar términos cuidadosamente).

# Apéndice D

---

## Código

---

Este apéndice contiene el código original empleado en la demostración del lema 2. Se trata de una librería de cálculo simbólico especialmente diseñada para tratar el formalismo de matriz densidad híbrida. Todas las gráficas obtenidas son aplicación directa de funciones aquí contenidas.

```
1 | from sympy import *
2 | from sympy.physics.quantum import *
3 | from sympy.physics.matrices import *
4 |
5 | herm_base = [eye(2), msigma(1), msigma(2), msigma(3)]
6 | herm_base_element_dim = sqrt(len(herm_base))
7 |
8 | def matrix_to_base(matrix):
9 |     """
10 |         Explanation
11 | =====
12 |         Returns the matrix coordinates in the chosen basis.
13 |
14 |     Parameters
15 | =====
16 |     """
17 |     if (matrix != Dagger(matrix)):
18 |         print('Matrix is not hermitian.')
19 |     base_coords = [(Rational(1/2)*(matrix*base_element) \
20 | .trace()).simplify() for base_element in herm_base]
21 |     return base_coords
22 |
23 | def base_to_matrix(base_coords):
24 |     """
25 |         Explanation
26 | =====
27 |         Returns the matrix representation given the coordinates.
28 |
29 |     Parameters
30 | =====
31 |     """
32 |     if (len(base_coords) != len(herm_base)):
33 |         raise ValueError('Incoherent array length.')
34 |     for coord in base_coords:
35 |         if(coord != conjugate(coord)):
36 |             print('Coordinates must be real numbers.')
37 |             break
38 |     matrix = zeros(herm_base_element_dim)
```

```

39     for i in range(len(herm_base)):
40         matrix = Add(matrix, base_coords[i]*herm_base[i])
41     return matrix
42
43 def base_to_bloch(base_coords):
44     """
45     Explanation
46     =====
47     Returns the Bloch coordinates of the object given its basis
48     coordinates (if the trace is 1).
49
50     Parameters
51     =====
52     """
53     if (len(base_coords) != len(herm_base)):
54         raise ValueError('Incoherent array length.')
55     for coord in base_coords:
56         if(coord != conjugate(coord)):
57             print('Coordinates must be real numbers.')
58             break
59     if (2*base_coords[0] != 1):
60         raise ValueError('Trace is not 1.')
61     bloch_coords = [2*base_coords[i] for i in range(1, \
62     len(base_coords))]
63     return bloch_coords
64
65 def bloch_to_base(bloch_coords):
66     """
67     Explanation
68     =====
69     Returns the basis coordinates of the object given
70     its Bloch coordinates (trace is assumed 1 for the
71     Bloch coordinates to be meaningful).
72
73     Parameters
74     =====
75     """
76     base_coords = [Rational(1/2), *[Rational(1/2)* \
77     bloch_coords_element for bloch_coords_element \
78     in bloch_coords]]
79     return base_coords
80
81 class Herm:
82     """
83     Explanation
84     =====
85     Encodes the information of an hermitian operator.
86     self.matrix: the matrix representation.
87     self.base_coords: the coordinates in the given basis.
88     self.bloch_object: boolean, true if and only if the
89     operator's trace is 1.
90     self.bloch_coordinates: only defined if self.bloch_object
91     is true. Operator's Bloch coordinates.
92     """
93     def __init__(self, *args):
94
95         if len(args)==1:
96             self.matrix = args[0]

```

```

97         self.base_coords = matrix_to_base(self.matrix)
98         self.bloch_object = (2*self.base_coords[0] \
99             .simplify() == 1)
100        if self.bloch_object:
101            self.bloch_coords = base_to_bloch( \
102                self.base_coords)
103        elif len(args)==len(herm_base):
104            self.base_coords = args
105            self.matrix = base_to_matrix(args)
106            self.bloch_object = (2*self.base_coords[0] \
107                .simplify() == 1)
108            if self.bloch_object:
109                self.bloch_coords = base_to_bloch( \
110                    self.base_coords)
111        else:
112            raise ValueError('Invalid number of input \
113                parameters.')
114
115    def marginal_conditional(herm):
116        marginal = 2*herm.base_coords[0]
117        conditional = Herm((herm.matrix/marginal) \
118            .applyfunc(simplify))
119        return marginal, conditional
120
121    class HybridDM:
122        """
123        Explanation
124        =====
125        Encodes the information of a hybrid density matrix.
126
127        As it is an hermitian operator, it has:
128        self.matrix: the matrix representation.
129        self.base_coords: the coordinates in the given basis.
130        self.bloch_object: boolean, true if and only if
131        the operator's trace is 1.
132        self.bloch_coordinates: only defined if
133        self.bloch_object is true. Operator's Bloch coordinates.
134
135        Moreover, the factorization is encoded in:
136        self.marginal: the marginal probability
137        density distribution.
138        self.conditional: the density matrix, a class Herm object.
139
140        To show how the representation chosen
141        determines the evolution:
142        self.convex_combination[k]: a list of two sets
143        of projectors and weights.
144        self.derivative: the time derivative using each
145        former convex_combination.
146
147        Initialization parameters
148        =====
149        herm: an hermitian operator, class Herm object.
150        """
151    def __init__(self, herm):
152        self.matrix = herm.matrix
153        self.base_coords = herm.base_coords
154        self.bloch_object = herm.bloch_object

```

```

155     if self.bloch_object:
156         self.bloch_coords = herm.bloch_coords
157         self.marginal, self.conditional = \
158             marginal_conditional(herm)
159         self.convex_combination = [None]*2
160         self.derivative = [None]*2
161
162     def factorization(self, theta, phi):
163         """
164         Explanation
165         =====
166         Returns two projectors and its weights,
167         given only one of them,
168         and whose weighted linear combination
169         returns the same density matrix.
170
171         Parameters
172         =====
173         Two angles on the Bloch sphere, which
174         represent an arbitrary projector.
175         """
176
177         rho = Matrix(self.conditional.bloch_coords)
178         pi_input = Matrix([sin(theta)*cos(phi), \
179             sin(theta)*sin(phi),cos(theta)])
180         x = pi_input - rho
181         lamda = (rho.norm()**2-1)/(rho.norm()**2 - \
182             2*(rho.transpose()*pi_input)[0]+1)
183         pi_output = rho + lamda*x
184
185         cweight2 = (pi_input - rho).norm()/(pi_input - \
186             pi_output).norm()
187         cweight1 = (pi_output - rho).norm()/(pi_input - \
188             pi_output).norm()
189
190         Pi_input = Herm(*bloch_to_base(pi_input \
191             .transpose().tolist()[0]))
192         Pi_output = Herm(*bloch_to_base(pi_output \
193             .transpose().tolist()[0]))
194
195         return Pi_input, Pi_output, cweight1, cweight2
196
197     def set_factorization(self, Pi_1, Pi_2, weight1, \
198         weight2, k):
199         """
200         Explanation
201         =====
202         Manually sets the values of the variables
203         at self.convex_combination[k].
204
205         Parameters
206         =====
207         Two angles on the Bloch sphere, which represent
208         an arbitrary projector.
209         k: convex_combination index.
210         """
211         if (k != 0 and k != 1):
212             raise ValueError('k must be 0 or 1.')
213             self.convex_combination[k] = [Pi_1, Pi_2, \

```

```

213     weight1, weight2]
214
215     def update_factorization(self, theta, phi, k, s):
216         """
217             Explanation
218             =====
219             Changes the values of the variables at
220             self.convex_combination[k].
221             s is a boolean to decide whether to apply
222             or not simplify.
223
224             Parameters
225             =====
226             Two angles on the Bloch sphere, which
227             represent an arbitrary projector.
228             k: convex_combination index.
229             s: boolean.
230             """
231
232         if (k != 0 and k != 1):
233             raise ValueError('k must be 0 or 1.')
234         if (s != 0 and s != 1):
235             raise ValueError('s must be 0 or 1.')
236
237         if s:
238             Pi_input, Pi_output, cweight1, \
239             cweight2 = self.factorization(theta, phi)
240             Pi_1 = Herm(Pi_input.matrix.applyfunc(simplify))
241             Pi_2 = Herm(Pi_output.matrix.applyfunc(simplify))
242             weight1 = (self.marginal*cweight1).simplify()
243             weight2 = (self.marginal*cweight2).simplify()
244             self.convex_combination[k] = [Pi_1, \
245             Pi_2, weight1, weight2]
246
247         else:
248             Pi_input, Pi_output, cweight1, \
249             cweight2 = self.factorization(theta, phi)
250             weight1 = self.marginal*cweight1
251             weight2 = self.marginal*cweight2
252             self.convex_combination[k] = \
253             [Pi_input, Pi_output, weight1, weight2]
254
255     def update_derivative(self, H, k, var):
256         """
257             Explanation
258             =====
259             Given a particular factorization (convex_combination[k]),
260             calculates the time derivative of
261             the hybrid density matrix
262             induced by a hybrid hamiltonian.
263
264             Parameters
265             =====
266             H: hamiltonian, class Herm object.
267             k: convex_combination index.
268             var: list of symbols.
269             """
270         if (k != 0 and k != 1):
271             raise ValueError('k must be 0 or 1.')

```

```

271
272     def pbracket(x,y):
273         return x.diff(q)*y.diff(p) - x.diff(p)*y.diff(q)
274
275     q = var[0]
276     p = var[1]
277
278     def element_pbracket_q(x):
279         return pbracket(x,q)
280
281     def element_pbracket_p(x):
282         return pbracket(x,p)
283
284     def matrix_pbracket(A,B):
285         dq = lambda x : x.diff(q)
286         dp = lambda x : x.diff(p)
287         return A.applyfunc(dq)*B.applyfunc(dp) - \
288             A.applyfunc(dp)*B.applyfunc(dq)
289
290         [Pi_1, Pi_2, weight1, weight2] = \
291         self.convex_combination[k]
292         Pi = [Pi_1.matrix, Pi_2.matrix]
293         weight = [weight1, weight2]
294
295         dq = lambda x : x.diff(q)
296         dp = lambda x : x.diff(p)
297
298         Term1 = -(I/hbar)*(H.matrix*self.matrix - \
299             self.matrix*H.matrix)
300
301         Term2 = zeros(2,2)
302         Term3 = zeros(2,2)
303
304         for j in range(len(Pi)):
305             Term2 = Term2 + (weight[j]*(
306                 ((Pi[j]*((H.matrix).applyfunc(
307                     element_pbracket_q))).trace())*Pi[j].applyfunc(dq) \
308                 + ((Pi[j]*((H.matrix).applyfunc(
309                     element_pbracket_p))).trace())*Pi[j].applyfunc(dp)))
310
311         for j in range(len(Pi)):
312             Term3 = Term3 + (matrix_pbracket(H.matrix,
313                 weight[j]*Pi[j])).trace()*Pi[j]
314
315         self.derivative[k] = Term1 + Term2 + Term3

```